INTT cluster errors issue

Tony Frawley
FSU

sPHENIX Simulations Meeting
3/21/2017

The problem

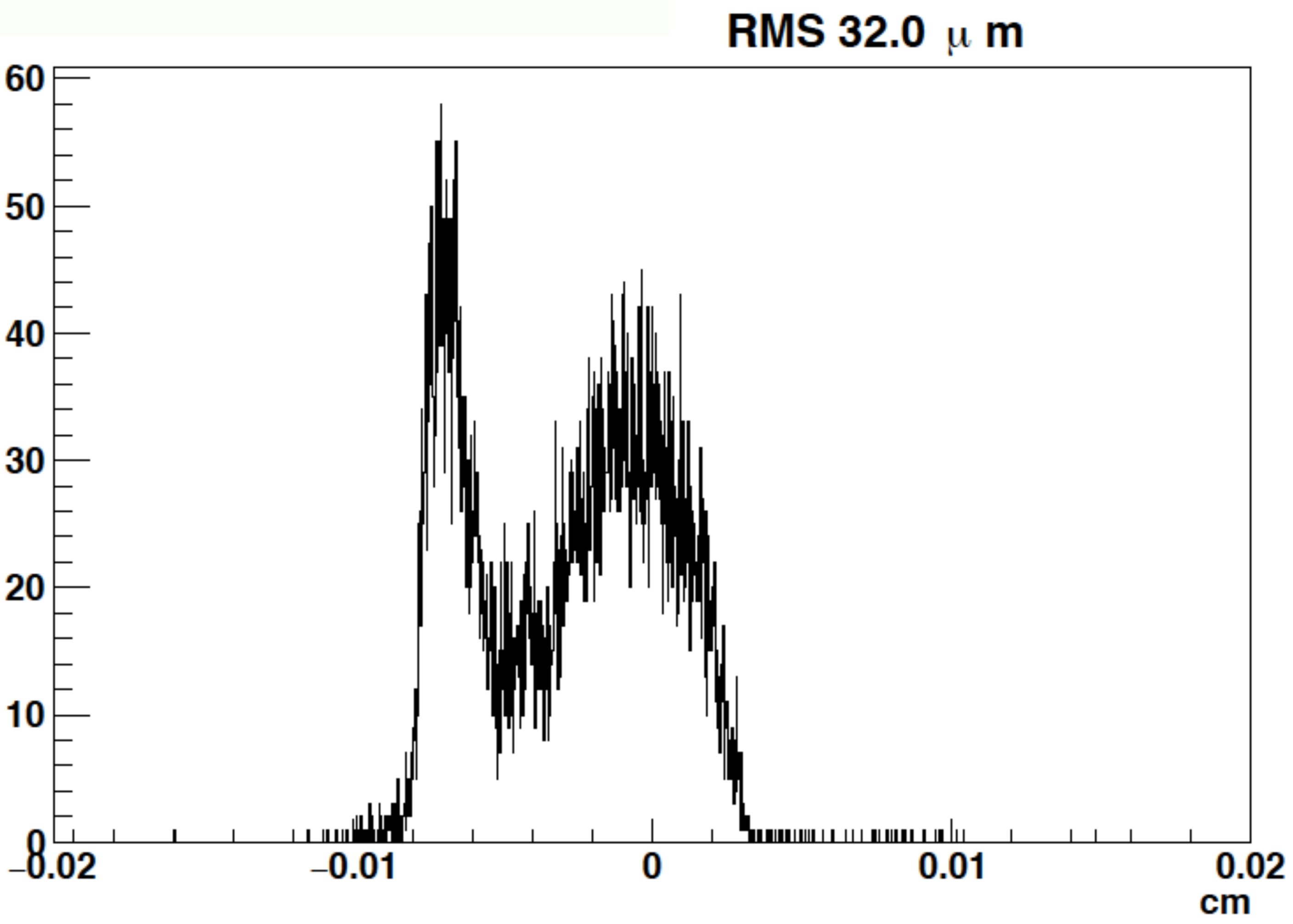
When running tracking simulations with the ladder implementation of the INTT, we see the following distribution of cluster errors relative to the truth hit position.

The left hand peak is low by one strip width.

Why is this?

A clue is that the clusters are essentially ALL single strip clusters.

That is wrong: with the ladder tilt of ~ 140 there should be ~ 30% clusters with 2 strips.



The cause - 1

I tracked the problem down to an issue in PHG4SiliconTrackerSteppingAction.

But I need to review a bit before describing that.

The ladder model is created in PHG4SiliconTrackerDetector
The silicon strips are placed in the sensors using the following commands:

```
  G4VPVParameterisation *stripparam = new PHG4SiliconTrackerStripParameterisation(nstrips_phi_cell * 2,
nstrips_z_sensor, strip_y * 2., strip_z * 2.);
   new G4PVParameterised(boost::str(boost::format("siactive_%d_%d") %sphxlayer %itype).c_str(), strip_volume,
siactive_volume, kZAxis, nstrips_phi_cell * 2 * nstrips_z_sensor, stripparam, false); // overlap check too long.
```

where G4PVParameterised places the array of identical strips in the sensor volume at the locations specified in the class PHG4SiliconTrackerStripParameterisation.

These silicon strips are then made the active volumes.

The cause - 2

The information for the active volumes (the hit strips) is extracted in PHG4SiliconTrackerUserSteppingAction. The relevant steps for us are:

```
G4TouchableHandle touch = aStep->GetPreStepPoint()->GetTouchableHandle();
G4VPhysicalVolume* volume = touch->GetVolume();

    // This just regurgitates the values set in PHG4SiliconTrackerParameterization
    // when the G4PVParameterized was defined
    G4ThreeVector strip_pos = volume->GetTranslation();

    strip_y_index = 0;
     for (int i=0; i<2*nstrips_phi_cell; ++i)
       {
         const double ymin = 2.*strip_y*(double)(i)   - 2.*strip_y*(double)nstrips_phi_cell;
         const double ymax = 2.*strip_y*(double)(i+1) - 2.*strip_y*(double)nstrips_phi_cell;
         if (strip_pos.y()/CLHEP::mm>ymin && strip_pos.y()/CLHEP::mm<=ymax)
           {
             strip_y_index = i;
           }
       }
```

where the calculations of strip boundaries match the strip placement parameterization. When the track crosses only one strip, this works well.

The cause - 3

The problem occurs when the track passes through two adjacent strips. In that case, PHG4SiliconTrackerUserSteppingAction is called for each active volume (strip) traversed by the track, and the code on the previous slide is thus called for each of the hit strips.

The behavior of the code is most easily traced by throwing single pions in the simulation, so that there is only one track. The cases where two strips are hit are obvious - there are two calls to the SteppingAction, and in world coordinates the exit point of the first call is identical to the entry point of the second call.

The problem is that in both calls to the SteppingAction the call

```
    G4ThreeVector strip_pos = volume->GetTranslation();
```

returns the strip position for the second hit strip. If the copy number of the strip is also requested, the copy number for both calls is also for the second hit strip. So the TouchableHandle returned by the call to aStep (see previous slide) points to the last strip crossed by the track, not the strip corresponding to the step for which the SteppingAction was called.

The cause - 3

Example of hits in two adjacent strips (Red: exit location of first strip = entry location of second strip):

```
PHG4SilicoTrackerSteppingActionfor siactive_4_0 volume->GetTranslation (0,-1.247,-56) volume->GetCopyNo() 904
   found strip y index = 113 (ymin -1.29 ymax -1.204)
   sphxlayer 4 inttlayer 1 ladderz 2 ladderphi 12 strip_z_index 0 strip_y_index 113
   location of strip center: strip_pos.x 0 strip_pos.y -1.247 strip_pos.z -56
G4 step information:
    world entry/exit point position: X -7.76586/-7.77726  Y 1.79395/1.79646  Z 1.47829/1.48045
    entry point local position in sensor: 0.1 -1.31546 -50.2971
   entry point local position in strip: 0.1 0.0175404 5.70294
```

Blue: This sensor y position is in strip (iy = 112, iz=0, copy = 896, y = -1.333) - GetCopyNo() returned wrong copy

```
PHG4SilicoTrackerSteppingAction for siactive_4_0 volume->GetTranslation (0,-1.247,-56) volume->GetCopyNo() 904
    found strip y index = 113 (ymin -1.29 ymax -1.204)
    sphxlayer 4 inttlayer 1 ladderz 2 ladderphi 12 strip_z_index 0 strip_y_index 113
    location of strip center: strip_pos.x 0 strip_pos.y -1.247 strip_pos.z -56
G4 step information:
    world entry/exit location: X -7.77726/-7.78588  Y 1.79646/1.79836  Z 1.48045/1.48208
    entry point local position in sensor: -0.0138329 -1.29 -50.2755
    entry point local position in strip: -0.0138329 -0.043 5.72452
```

Blue: This sensor y position is in strip (iy = 113, iz = 0, copy = 904, y = -1.247) - GetCopyNo() returned correct copy

How do I fix this?

Any suggestions on how to fix this?

My look at the documentation on G4PVParameterised turned up nothing.

Is there a different way to navigate the to the volume than asking the stepPoint for the volume pointer?

If all else fails, we could use an assembly volume for the placement instead of using the G4PVParameterised to place the strips.